

AD-A075 575

STANFORD UNIV CA DEPT OF OPERATIONS RESEARCH

F/G 12/1

A BOUNDING TECHNIQUE FOR INTEGER LINEAR PROGRAMMING WITH BINARY--ETC(U)

JUN 79 F S HILLIER , N E JACOMIN

N00014-76-C-0418

UNCLASSIFIED

TR-86

NL

1 OF 1  
AD-A075575



END  
DATE  
FILMED

11-79

DDC

# LEVEL<sup>II</sup>

12

A BOUNDING TECHNIQUE FOR INTEGER LINEAR PROGRAMMING  
WITH BINARY VARIABLES

BY

FREDERICK S. HILLIER and NANCY EILEEN JACQUIN

TECHNICAL REPORT NO. 86

JUNE 1979

PREPARED UNDER CONTRACT

N00014-76-C-0418 (NR-047-061)

FOR THE OFFICE OF NAVAL RESEARCH

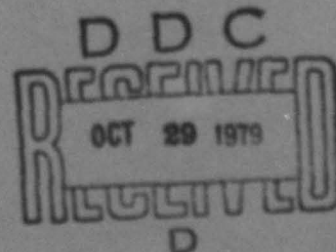
Frederick S. Hillier, Project Director

Reproduction in Whole or in Part is Permitted  
for any Purpose of the United States Government

This document has been approved for public release  
and sale; its distribution is unlimited.

DDC FILE COPY

DEPARTMENT OF OPERATIONS RESEARCH  
STANFORD UNIVERSITY  
STANFORD, CALIFORNIA



79 10 26 048

AD A 0 75575

6 A BOUNDING TECHNIQUE FOR INTEGER LINEAR PROGRAMMING  
WITH BINARY VARIABLES

BY

10 FREDERICK S. HILLIER and NANCY EILEEN JACOMIN

9 TECHNICAL REPORT NO. 86

11 JUNE 1979

12 17

15 PREPARED UNDER CONTRACT  
N00014-76-C-0418 (NR-047-061)  
14 TR-86  
NSF-MCS 76-81259  
FOR THE OFFICE OF NAVAL RESEARCH

Frederick S. Hillier, Project Director

Reproduction in Whole or in Part is Permitted  
for any purpose of the United States Government

This document has been approved for public release  
and sale; its distribution is unlimited.

This research was supported in part by National Science Foundation  
Grant NSF/SOL No. MCS 76-81259 A01, Dept. of Operations Research  
Stanford University.

DEPARTMENT OF OPERATIONS RESEARCH  
STANFORD UNIVERSITY  
STANFORD, CALIFORNIA

DDC  
RECEIVED  
OCT 29 1979  
D

Accession For	
NTIS GRA&I	
DDC TAB	
Unannounced	
Justification	
By	
Distribution/	
Availability Codes	
Dist..	Avail and/or special
A	

402 766

mt

## 1. Introduction

This paper presents a bounding technique for use in implicit enumeration algorithms for solving the integer linear programming problem with binary variables. This problem may be stated as follows,

$$\text{maximize } x_0 = \sum_{j=1}^n c_j x_j,$$

subject to

$$(P) \quad \sum_{j=1}^n a_{ij} x_j \leq b_i, \quad (i = 1, 2, \dots, m)$$

$$x_j = 0 \text{ or } 1, \quad (j = 1, 2, \dots, n).$$

The main assumptions used by this technique are (1) the linear programming relaxation of (P),

$$\text{maximize } x_0 = \sum_{j=1}^n c_j x_j,$$

subject to

$$(P_R) \quad \sum_{j=1}^n a_{ij} x_j \leq b_i, \quad (i = 1, 2, \dots, m)$$

$$0 \leq x_j \leq 1, \quad (j = 1, 2, \dots, n)$$

possesses a unique optimal solution, (2) the optimal solution to  $(P_R)$  is not binary, and (3) a good feasible solution to (P) is available. If (2) does not hold then the optimal solution to  $(P_R)$  is also the optimal solution to (P) and the algorithm would not be required. Assumption (3) may be met by using one of the many heuristic procedures in the literature [1,4,6,7]. An alternative to assumption (3) requires that a lower bound on the optimal objective function value of a related problem  $(P')$ , defined in Section 3, be available.

In what follows we show that "joint" bounds can be obtained on the values of a subset of the variables by adapting the approach used by the Bound-and-Scan algorithm [3] in which the variables are restricted to be non-negative integers rather than binary. In addition we give good indications that these bounds may

be used efficiently. Finally, a class of problems particularly well-suited to this bounding procedure is specified.

## 2. Conceptual Outline of the Bounding Technique

This technique is motivated by the following considerations. In an implicit enumeration algorithm the goal is to quickly eliminate large subsets of the  $2^n$  possible combinations of values for the variables by various tests. Having a bound on the objective function value of the optimal solution provides one such test since constraining feasible points to yield a better objective function value greatly decreases the size of the feasible region. Also, this new constraint assures that any new feasible solution will decrease the size of the feasible region even further. Moreover, if we can obtain a representation for a group of variables that (1) allows us to fix the values of some of the variables in an optimal solution to (P) and (2) yields "joint" bounds on them which allow systematic generation of all relevant combinations of their values, we will have eliminated even more of the  $2^n$  combinations. Further, if the number of variables in this group is large as a function of  $n$  and we use the "joint" bounds in a computationally frugal way, additional time savings will be achieved as a result of having only a small number of variables to deal with in another manner. The technique presented here seeks to take advantage of these ideas.

Consider the region determined by the constraints that are binding<sup>1</sup> in the optimal solution to  $(P_R)$ , after making a change of variables  $(x'_j = 1 - x_j)$  for those variables,  $x_j$ , whose complements,  $(1 - x_j)$ , were nonbasic in this solution [5, pp. 659-662]. In addition, require that the value of the objective function

---

<sup>1</sup>Here we exclude constraints that are binding due to degeneracy. Hence, the binding constraints are those constraints  $\sum_{j=1}^n a_{ij}x_j \leq b_i$  ( $i = 1, 2, \dots, m$ ) whose slack variables are nonbasic in the optimal solution to  $(P_R)$ , and those non-negativity constraints that correspond, after the change of variables, to nonbasic variables in this solution.

not be less than the bound generated by the known feasible solution to (P). The optimal solution to (P) clearly lies in this region. Identify the extreme points of this region, and represent each feasible point (as defined by this region) as a convex combination of these extreme points; then use the remaining constraints to perhaps eliminate some of these convex combinations from further consideration. This provides a convenient way of identifying the relevant combinations of values for the variables that were nonbasic, at least after the change of variables, in the optimal solution to  $(P_R)$ . Each time a relevant set of values is determined we exit from this technique to the underlying algorithm to see if this partial set of values for the  $x_j$  can yield a completion which satisfies all the constraints of (P). If we find such a feasible solution it yields a new lower bound on the optimal objective function value of (P). We then seek to identify another relevant set of values for this same group of variables. Hence, an iteration corresponds to a relevant set of values for these variables and the algorithm terminates when all such sets have been considered.

### 3. Notation and Initialization

This section introduces the notation and terminology that will be used in what follows. An alternative representation of (P) is

$$\begin{aligned} &\text{maximize} \quad x_0 = \underline{c} \underline{x} \\ &\text{subject to} \\ &\quad \underline{A} \underline{x} \leq \underline{b} \\ &\quad x_j = 0 \text{ or } 1, \quad (j = 1, \dots, n) \end{aligned}$$

where  $\underline{c}$  is a  $1 \times n$  vector,  $\underline{x}$  is an  $n \times 1$  vector,  $\underline{b}$  is an  $m \times 1$  vector and  $A$  is an  $m \times n$  matrix. An  $n \times 1$  vector  $\underline{x}$  is a feasible solution to (P) if it is in the region defined by the constraints; it is an optimal

solution if it is feasible and maximizes the objective function value among all feasible solutions. The value of  $x_0$  for an optimal solution will be called the optimal objective function value. If values have been assigned to only certain of the variables (components of  $\underline{x}$ ), this specification of values will be called a partial solution for these variables. Given a partial solution, a "completion" is a solution resulting from specification of values for the remaining variables. An eligible partial solution is one that does not violate any known bounds on the variables involved, i.e., one that, based on our present information, could yield a completion that is an optimal solution.

The optimal solution to  $(P_R)$  may be included in the card input stream or  $(P_R)$  may be solved during initialization if the underlying algorithm is simplex-based. In either case, we will denote this solution as  $\underline{x}^* = [x_1^*, \dots, x_n^*]^T$  and its objective function value as  $x_0^* = \underline{c} \underline{x}^*$ . Also we define

$$J_B = \{j | x_j^* \text{ is basic}\},$$

$$J_0 = \{j | x_j^* = 0 \text{ and } x_j^* \text{ is nonbasic}\},$$

$$J_1 = \{j | x_j^* = 1 \text{ and } 1 - x_j^* \text{ is nonbasic}\},$$

$$J_N = J_0 \cup J_1,$$

$$K_B = \{i | i \in \{1, \dots, m\} \text{ and the } i\text{th constraint is binding at } \underline{x}^*\},$$

$$K_N = \{1, \dots, m\} - K_B.$$

We wish to make the following change of variables to define  $\underline{x}' \in \mathbb{R}^n$  for any  $\underline{x} \in \mathbb{R}^n$ .

$$x'_j = \begin{cases} 1 - x_j, & \text{if } j \in J_1 \\ x_j, & \text{otherwise.} \end{cases}$$

This change of variables directly induces the problem

$$\text{maximize } x'_0 = \underline{c}' \underline{x}'$$

subject to

$(P')$

$$\underline{A}' \underline{x}' \leq \underline{b}'$$

$$x'_j = 0 \text{ or } 1$$

from (P). Note that an optimal solution to (P'),  $\underline{x}'^{opt}$ , yields an optimal solution,  $\underline{x}^{opt}$ , to (P) via

$$x_j^{opt} = \begin{cases} 1 - x_j'^{opt}, & \text{if } j \in J_1 \\ x_j'^{opt}, & \text{otherwise.} \end{cases}$$

Henceforth we will direct our attention to solving (P').

For notational convenience it will be assumed that the original indexing of the  $x_j$  and  $b_i$  was such that if  $n_B = |J_B|$ , then

$$\begin{aligned} J_B &= \{1, \dots, n_B\}, \\ K_B &= \{1, \dots, n_B\}, \\ J_N &= \{n_B+1, \dots, n\}, \\ K_N &= \{n_B+1, \dots, m\}. \end{aligned}$$

Finally denote the initial good feasible solution to (P) by  $\underline{x}^{(F)} = [x_1^F, \dots, x_n^F]^T$ . Now define  $\underline{x}'^{(F)}$  by

$$x_j'^{(F)} = \begin{cases} 1 - x_j^{(F)}, & \text{if } j \in J_1 \\ x_j^{(F)}, & \text{otherwise,} \end{cases}$$

and let  $x_0'^{(F)} = \underline{c}' \underline{x}'^{(F)}$ . Note that  $x_0'^{(F)}$  is a lower bound on the optimal objective function value of (P').

#### 4. Technique for Bounding $\{x_j | j \in J_N\}$

Define the  $n \times n$  matrix  $A^*$  whose  $i$ th row is

$$\underline{A}_i^* = \begin{cases} \underline{A}_i', & i = 1, \dots, n_B \\ -\underline{e}_i, & i = n_B+1, \dots, n, \end{cases}$$

where  $\underline{e}_i$  is the unit  $n$ -vector with unity assigned to component  $i$ . Also define  $\underline{b}^* \in \mathbb{R}^n$  whose  $i$ th component is

$$b_i^* = \begin{cases} b_i', & i = 1, \dots, n_B \\ 0, & i = n_B + 1, \dots, n. \end{cases}$$

Now consider the set of points satisfying

$$\begin{aligned} (i) \quad & A^* \underline{x} \leq \underline{b}^*, \\ (ii) \quad & \underline{c}' \underline{x} \geq x_0'^{(F)}. \end{aligned}$$

Since (i) is the set of constraints that are binding on  $\underline{x}^*$  (induced by  $\underline{x}^*$  via the change of variables) and  $x_0'^{(F)}$  is a lower bound on the optimal objective function value of (P'), any optimal solution to (P') must be in this set. If  $(\underline{c}' \underline{x}^* - \underline{c}' \underline{x}'^{(F)})$  is small, this set will contain few binary points and the search for an optimal binary point may not be difficult.

We now seek to find the  $(n+1)$  extreme points of the  $n$ -simplex defined by (i) and (ii). The first  $n$  extreme points are determined by the system of equations,

$$A^{(i)} \underline{x}^{(i)} = \underline{b}^{(i)}, \quad (i = 1, \dots, n)$$

where  $A^{(i)}$  is obtained by replacing row  $i$  of  $A^*$  by  $\underline{c}'$ , and  $\underline{b}^{(i)}$  is the vector obtained by replacing component  $i$  of  $\underline{b}^*$  by  $x_0'^{(F)}$ . The  $(n+1)^{st}$  extreme point is  $\underline{x}^{(0)} = \underline{x}^*$  [3, p. 646]. It can be shown that  $A^{(i)}$  is nonsingular  $(i = 1, \dots, n)$  [3, p. 646] and, hence, we have explicitly

$$\underline{x}^{(i)} = (A^{(i)})^{-1} \underline{b}^{(i)}, \quad (i = 1, \dots, n).$$

We wish to represent the feasible points as convex combinations of these extreme points. To this end define

$$M = \begin{bmatrix} \underline{x}^{(0)} & \underline{x}^{(1)} & \dots & \underline{x}^{(n)} \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

and

$$\underline{\rho} = [\rho_0, \dots, \rho_n]^T.$$

Then setting

$$\underline{M}_0 = \begin{bmatrix} \underline{x} \\ 1 \end{bmatrix}$$

yields such a representation for  $\underline{x}$  via  $\underline{\rho}$ .  $\underline{x}$  is in the simplex determined by (i) and (ii) if and only if  $\underline{\rho} \geq 0$ . For an arbitrary  $\underline{x}$ ,  $\underline{\rho}$  exists by virtue of the fact that  $M$  is nonsingular [3, pp. 648,649], so

$$\underline{\rho} = M^{-1} \begin{bmatrix} \underline{x} \\ 1 \end{bmatrix}$$

provides the relevant vector  $\underline{\rho}$ .

Fortunately, this representation can be obtained directly without explicitly performing a matrix inversion, since  $\underline{A}_i^* \underline{x}^{(j)} = b_i^*$  for  $j \neq i$  (by definition of  $A^{(i)}$  and  $b^{(i)}$ ,  $i = 1, \dots, n$ ), so

$$\begin{aligned} b_i^* - \underline{A}_i^* \underline{x} &= b_i^* - \sum_{j=0}^n \rho_j \underline{A}_i^* \underline{x}^{(j)} \\ &= b_i^* - (1 - \rho_i) b_i^* - \rho_i \underline{A}_i^* \underline{x}^{(i)} \\ &= \rho_i (b_i^* - \underline{A}_i^* \underline{x}^{(i)}). \end{aligned}$$

Therefore,

$$\rho_i = \frac{b_i^* - \underline{A}_i^* \underline{x}}{b_i^* - \underline{A}_i^* \underline{x}^{(i)}}, \quad i = 1, \dots, n$$

and, similarly,

$$\rho_0 = \frac{\underline{c}' \underline{x} - x_0'(F)}{\underline{c}' \underline{x}^{(0)} - x_0'(F)}.$$

Solving (P') has now been reduced to finding those  $\underline{x}$  with  $\underline{\rho} \geq 0$  which satisfy the constraints of (P') that are not binding on  $\underline{x}^*$  and identifying the one which maximizes  $\underline{c}' \underline{x}$ .

It can be shown that each element of  $\{\underline{x}_j | j \in J_N\}$  has the property that it has value zero at all of the extreme points of our  $n$ -simplex except one, where

it has a strictly positive value. In particular,  $x_j^{(j)} > 0$  for each  $j \in J_N$ . This follows from Lemma 3 [3, p. 646], which for completeness is restated here.

Lemma: Under the assumptions of Section 1,  $\underline{x}^{(i)}$  ( $i = 0, \dots, n$ ) satisfies constraints (i) and (ii). Furthermore,  $\underline{A}_i^* \underline{x}^{(i)} < \underline{b}_i^*$  for  $i = 1, \dots, n$  and  $\underline{c}^* \underline{x}^{(0)} > x_0'(F)$ .

Hence, for  $j \in J_N$ ,

$$\rho_j = \frac{b_j^* - \underline{A}_j^* \underline{x}}{b_j^* - \underline{A}_j^* \underline{x}^{(j)}} = \frac{x_j}{x_j^{(j)}}.$$

This result, combined with the fact that  $\rho_0$  increases as the best known objective function value of (P') increases, yields the following theorem [3, p. 652].

Theorem: A necessary condition for  $[x_{n_B+1}, \dots, x_n]^T$  to be an eligible partial solution to (P') for the variables whose indices are in  $J_N$  is that

$$\sum_{j=n_B+1}^n \frac{x_j}{x_j^{(j)}} \leq 1 - \rho_0^B,$$

where  $\rho_0^B$  is the value of  $\rho_0$  for the current best feasible solution to (P'),  $\underline{x}^B = [x_1^B, \dots, x_n^B]^T$  (with objective function value  $x_0^B$ ).

This leads immediately to the following corollary.

Corollary: If  $x_j^{(j)} < 1$  where  $j \in J_N$ , then  $x_j' = 0$  in any optimal solution to (P').

Proof:  $x_j^{(j)} < 1$  implies that  $1/x_j^{(j)} > 1$ , since  $x_j^{(j)} > 0$ . Hence,  $\rho_j > 1$  when  $x_j = 1$  and this partial solution for  $x_j$  is not eligible.

Thus we may set  $x_j' \equiv 0$  for those  $j$  where  $x_j^{(j)} < 1$ . For notational convenience let this occur for  $j = n_B + 1, \dots, n_0$ , and our necessary condition is now reduced to

$$\sum_{j=n_0+1}^n \frac{x_j}{x_j^{(j)}} \leq 1 - \rho_0^B.$$

Thus, by calculating  $1/x_j^{(j)}$  for  $j = n_0+1, \dots, n$  once at the outset and updating  $\rho_0$  whenever a new feasible solution to  $(P')$  is found, a single addition and comparison are needed to check whether a current partial solution for these variables remains eligible if a particular value is increased from zero to one. Hence, it is simple to generate all the eligible partial solutions one at a time, stopping each time to explore its completions using the underlying algorithm.

A possible enumeration scheme is initiated by starting at  $[x_{n_0+1}, \dots, x_n]^T = \underline{0}$ . Each subsequent partial solution is obtained by first searching the set  $\{x_{n_0+1}, \dots, x_n\}$  for the first zero value,  $x_q$ , and then setting  $x_{n_0+1} = x_{n_0+2} = \dots = x_{q-1} = 0$  and  $x_q = 1$  and checking the necessary condition for eligibility. If this new partial solution is not eligible, again search for the first zero value,  $x_p$ , and set  $x_{n_0+1} = \dots = x_{p-1} = 0$  and  $x_p = 1$  and check for eligibility, etc. When there does not exist a zero-valued variable in  $\{x_{n_0+1}, \dots, x_n\}$  the eligible partial solutions have been exhausted and the algorithm terminates.

### 5. Computational Considerations

The bounding technique presented in the preceding section has not yet been tested for computational efficiency. However, we discuss below some indications that it will be efficient, and then consider which kinds of problems can best take advantage of it.

It was mentioned in Section 4 that a small value of  $(\underline{c}'\underline{x}^* - \underline{c}'\underline{x}^{(F)})$  is a key to the efficiency of the technique. Several heuristic procedures for obtaining  $\underline{x}^{(F)}$  have been tested and shown to give good results [9], so a relatively small value of the above seems to be a reasonable expectation. Such

a value also tends to yield relatively small values for the  $x_j^{(j)}$ , which greatly reduces the number of partial solutions that are eligible. For example, if  $1 \leq x_j^{(j)} < 2$  for all  $j = n_0 + 1, \dots, n$ , then the number of eligible partial solutions grows only linearly with  $(n - n_0)$ , whereas this number could grow essentially exponentially if the  $x_j^{(j)}$  are sufficiently large.

In identifying the extreme points  $\underline{x}^{(1)}, \dots, \underline{x}^{(n)}$  it is not really necessary to find the  $(A^{(i)})^{-1}$  or even to construct the  $A^{(i)}$  (or, in fact,  $A^*$ ) in a simplex-based code. Beginning with the inverse of the final basis matrix for  $(P_R)$  (recalling that the change of variables does not affect the basic columns of  $A$ ), the extreme points may be computed by performing a succession of pivots. First append the constraint  $\underline{c}'\underline{x}' \geq x_0^{(F)}$  to  $(P_R')$  (induced by  $(P_R)$  via the change of variables), introducing its slack variable as the additional basic variable in the optimal solution. Now successively introduce each nonbasic variable  $(x_j : j \in J_N)$  into the basis while removing the variable that was introduced into the preceding basis. The first  $n$  components of the resulting  $n$  basic solutions are the desired extreme points.

An explicit representation for the  $\rho_i$  has already been given, obviating the need for finding  $M^{-1}$  or even constructing  $M$ .

Hence, though the theory involved in arriving at this approach is rather involved, the only additional storage (beyond that of the underlying algorithm) is  $\underline{x}^{(i)}$  ( $i = 0, \dots, n$ ),  $\rho_0^B$ ,  $1/x_j^{(j)}$  ( $j=n_0+1, \dots, n$ ),  $n_B$ , and a vector indicating which  $j$  are in  $J_1$ . The latter is needed to deal with the required change of variables. The added computation involves the change of variables, the addition of the constraint  $\underline{c}'\underline{x}' \geq x_0^{(F)}$  to  $(P_R')$ ,  $n$  pivots to find  $\underline{x}^{(1)}, \dots, \underline{x}^{(n)}$ , the computation of  $1/x_j^{(j)}$  ( $j=n_0+1, \dots, n$ ) and then one addition and comparison for each partial solution in the enumeration scheme delineated in the previous section.

These modest computational and storage requirements suggest that we should usually be able to efficiently investigate a relatively large number of variables with this bounding technique. In fact, a similar technique embedded in Hillier's Bound-and-Scan algorithm for the general integer linear programming problem has proven to be quite efficient under computational testing [3].

We know that  $n_B$ , the number of variables that are not handled by this technique, is bounded above by  $m$ , the number of functional constraints. This result follows from Weingartner's proof that at most  $m$  of the original variables can be basic in a solution to (P) [8, pp. 35-37]. Therefore, the bounding technique seems particularly promising for problems where  $m$  is small, since relatively few variables would then need to be handled by the underlying algorithm.

In summary, in a simplex-based implicit enumeration procedure to solve (P), the use of this bounding technique for problems where the number of constraints is small in comparison to the number of variables seems very promising computationally. It will be implemented and tested in the near future and the results reported.

#### References

1. Balas, Egon and Clarence H. Martin, "Pivot and Complement - A Heuristic for 0-1 Programming," Management Sciences Research Report No. 414, Carnegie-Mellon University (1978).
2. Garfinkel, Robert S. and George L. Nemhauser, Integer Programming, John Wiley & Sons, New York, 1972.
3. Hillier, Frederick S., "A Bound-and-Scan Algorithm for Pure Integer Linear Programming With General Variables," Operations Research 17, 638-679 (1969).
4. \_\_\_\_\_, "Efficient Heuristic Procedures for Integer Linear Programming with an Interior," Operations Research 17, 600-637 (1969).
5. Hillier, Frederick S. and Gerald J. Lieberman, Operations Research, Second Edition, Holden-Day, Inc., San Francisco, 1974.
6. Senju, Shizuo and Yoshiaki Toyoda, "An Approach to Linear Programming with 0-1 Variables," Management Science 15, B196-B207 (1968).
7. Toyoda, Yoshiaki, "A Simplified Algorithm for Obtaining Approximate Solutions to Zero-One Programming Problems," Management Science 21, 1417-1427 (1975).

8. Weingartner, H. Martin, Mathematical Programming and the Analysis of Capital Budgeting, Prentice-Hall, Englewood Cliffs, NJ, 1963.
9. Zanakis, Stelios H., "Heuristic 0-1 Linear Programming: An Experimental Comparison of Three Methods," Management Science 24, 91-104 (1977).

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 86	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A BOUNDING TECHNIQUE FOR INTEGER LINEAR PROGRAMMING WITH BINARY VARIABLES		5. TYPE OF REPORT & PERIOD COVERED TECHNICAL REPORT
7. AUTHOR(s) FREDERICK S. HILLIER and NANCY E. JACQMIN		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS DEPT. OF OPERATIONS RESEARCH STANFORD UNIVERSITY, STANFORD, CALIF.		8. CONTRACT OR GRANT NUMBER(s) N00014-76-C-0418
11. CONTROLLING OFFICE NAME AND ADDRESS OPERATIONS RESEARCH PROGRAM OFFICE OF NAVAL RESEARCH CODE 434 ARLINGTON, VA. 22217		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR-047-061
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE JUNE 1979
		13. NUMBER OF PAGES 12
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES This research was supported in part by National Science Foundation Grant NSF/SOL No. MCS 76-81259 A01, Dept. of Operations Research, Stanford University.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  INTEGER PROGRAMMING IMPLICIT ENUMERATION ALGORITHMS		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  <div style="text-align: center;">SEE OTHER SIDE</div>		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 68 IS OBSOLETE  
S/N 0102-010-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

A Bounding Technique for Integer Linear Programming  
with Binary Variables

by

Frederick S. Hillier and Nancy E. Jacqmin  
Department of Operations Research  
Stanford University

ABSTRACT

→ We present a bounding technique for use in implicit enumeration algorithms for solving the integer linear programming problem with binary variables. The main assumptions used by this technique are (1) the associated linear program, obtained by dropping the integrality constraints on the variables, possesses a unique optimal solution, (2) this optimal solution is not binary, and (3) a good feasible solution to the original problem is available. An alternative to <sup>the</sup> assumption (3) which is weaker is also presented. <sup>last</sup>

We show that ~~joint~~ bounds can be obtained on the values of a subset of the variables. In addition we give an efficient method to implement this bounding technique. Finally, a class of problems particularly well-suited to this bounding procedure is specified. ←

Report No. 86

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)